

.

Ivex.[s/u].[ms].[ds] rD, rA, rB

Γ

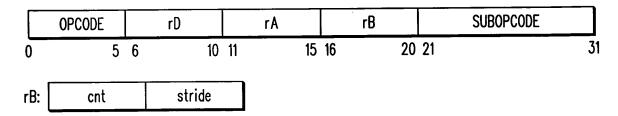


FIG. 2

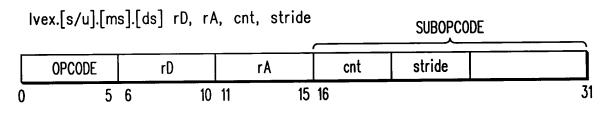


FIG. 3

Imvex.[s/u].[ms].[ds] rD, rA, rB

				T
OPCODE	rD	rA	rB	SUBOPCODE
	1			

rB: cnt stride skip skip_cnt

FIG. 4

Imvex.[s/u].[ms].[ds] rD, rA, cnt, stride, skip, skip_cnt

OPCODE rD rA cnt stride skip skip_cnt

Imvex2.[s/u].[ms].[ds] rD, rA, rB

Γ

	OPCODE	rD	rA	rB	SUBOPCODE
rB:	cnt	rcnt	stride	skip]

FIG. 6

Istrmvex.[s/u].[ms].[ds] rD, rA, rB

OPCODE	rD	rA	rB	SUBOPCODE

rB: cnt rcnt stride skip skip_cnt

FIG. 7

stvex.[s/u].[ms].[ss].[h/I] rS, rA, rB

OPCODE rS rA	rB	SUBOPCODE
--------------	----	-----------

rB: cnt stride

 $stmvex.[s/u].[ms].[ss].[h/I] \ rS, \ rA, \ rB$

 Γ

OPCODE	rS	rA	rB	SUBOPCODE
			-	

rB: cnt stride skip skip_cnt

FIG. 9

 $stmvex2.[s/u].[ms].[ss].[h/I] \ rS, \ rA, \ rB$

OPCODE rS rA rB SUBOPCODE

rB: cnt rcnt stride skip

FIG. 10

ststrmvex.[s/u].[ms].[ss].[h/I] rS, rA, rB

OPCODE rS rA rB	SUBOPCODE
-----------------	-----------

rB: cnt rcnt stride skip skip_cnt

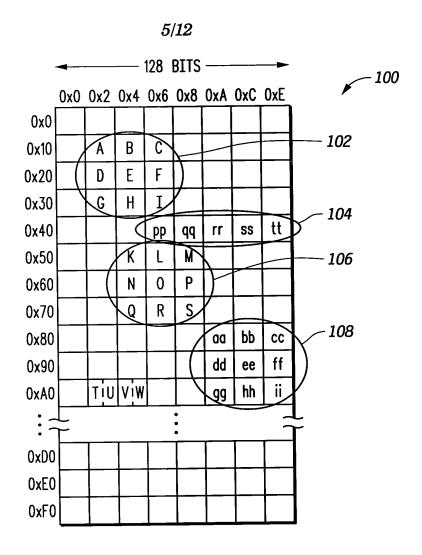


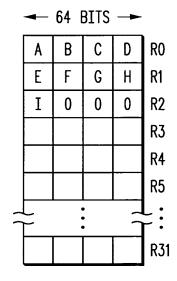
FIG. 12

← 64 BITS ←				-	64 E	BITS				
A	В	С	0	R0		D	E	F	0	R0
K	L	М	0	R1	•	N	0	Р	0	R1
A+K	B+L	C+M	0	R2		D+N	E+0	F+P	0	R2
				R3		G	Ή	I	0	R3
				R4		Q	R	S	0	R4
				R5		G+Q	H+R	I+S	0	R5
₹ T	<u> </u>	:	2	L:	=			•))	L :
				R31						R31
			_	_	~~			J	۔	4

FIG. 13

FIG. 14

6/12



Γ

	← 64 BITS →								
	A	В	С	0	R0				
	D	E	F	0	R1				
	G	H	Ι	0	R2				
					R3				
	+17	+įU	- iV	0	R4				
					R5				
())	<u> </u> :				
					R31				

FIG. 15

FIG. 16

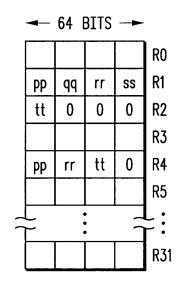
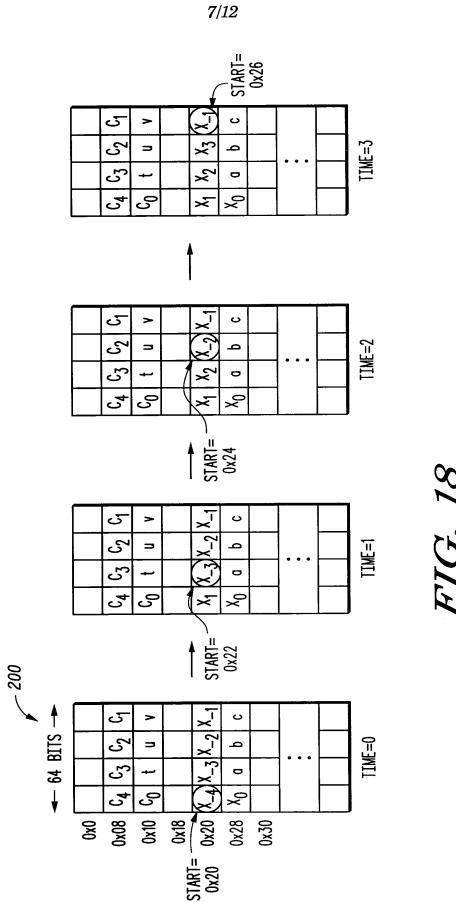


FIG. 17



→ 64 BITS →								
	0x08							
	0x20							
	:							
C ₄	C_4 C_3 C_2 C_1							
c ₀	0	0	c ₀	R7				
X ₋₄	X ₋₃	X ₋₂	X ₋₁	R8				
Х ₀	X ₀ 0 0 0							
$C_4 \cdot X_{-4}$	$C_4 \cdot X_{-4} + C_3 \cdot X_{-3} + C_2 \cdot X_{-2} + C_1 \cdot X_{-1}$							
c ⁰	• X ₀ +0 • (0+0-0+0	0-0	R11				

0x22								
:								
C_4 C_3 C_2 C_1								
c ₀	c_0 0 0 c_0							
X ₋₃	X_{-3} X_{-2} X_{-1} X_{0}							
X ₁	X ₁ 0 0 0							
$C_4 \cdot X_{-3} + C_3 \cdot X_{-2} + C_2 \cdot X_{-1} + C_1 \cdot X_0$								
c ⁰	•X ₁ +0•0)+0•0+0	•0	R11				

– 64 BITS —

FIG. 19

FIG. 20

← 64 BITS ← ►						
0x24						
:						
C_4 C_3 C_2 C_1						
C_0 0 0 C_0						
X_{-2} X_{-1} X_0 X_1						
X ₂ 0 0 0						
$c_4 \cdot x_{-2} + c_3 \cdot x_{-1} + c_2 \cdot x_0 + c_1 \cdot x_1$						
c ₀	•X ₂ +0•()+0•0+0	-0	R11		

-	— 64 I	BITS —	>				
	0x26						
:							
C ₄	c ₃	c ₂	C ₁	R6			
CO	0	0	c ₀	R7			
X ₋₁	x_0	X ₁	X ₂	R8			
X ₃	X ₃ 0 0 0						
$C_4 \cdot X_{-1} + C_3 \cdot X_0 + C_2 \cdot X_1 + C_1 \cdot X_2$							
C ₀ ·X ₃ +0·0+0·0+0·0							

FIG. 21

→ 64 BITS →							
Α	R1						
K	N	Q	0	R2			
0x12							
0x54							

Г

FIG. 23

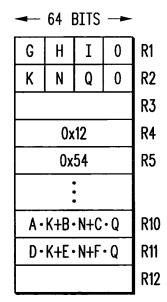


FIG. 25

—	← 64 BITS →							
D	Ε	F	0	R1				
K	N	Q	0	R2				
				R3				
	0x12							
	0x	54		R5				
	:							
A٠	A·K+B·N+C·Q							
	R11							
	R12							

FIG. 24

← 64 BITS ←								
G	H	I	0	R1				
L	L 0 R 0							
				R3				
	0x12							
	0x	54		R5				
	:							
A٠	R10							
D٠	R11							
G٠	G·K+H·N+I·Q							

lmvex_skip_once.[s/u].[ms].[ds] rD, rA, rB

Г

OPCODE rD rA rB SUBOPCODE	
---------------------------	--

rB: cnt stride skip skip_cnt

FIG. 27

 $lmvex_cb.[s/u].[ms].[ds] rD, rA, rB$

- 1					
	OPCODE	rD	rA	rB	SUBOPCODE

rB: buffer_size offset

FIG. 28

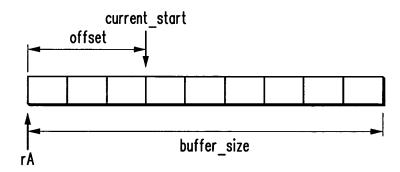


FIG. 29

lstrmvex_cb.[s/u].[ms].[ds] rD, rA, rB

OPCODE	rD	rA	rB	SUBOPCODE

rB: buffer_size offset

Imvex_fft.[s/u].[ms].[ds] rD, rA, rB

			,	
OPCODE	rD	rA	rB	SUBOPCODE

rB: radix

Γ

FIG. 31

stmvex_fft.[s/u].[ms].[ss] rS, rA, rB

OPCODE	rD	rA	rB	SUBOPCODE

rB: radix

FIG. 32

 $Imstrmvex_fft.[s/u].[ms].[ds] \ rD, \ rA, \ rB$

ODCODE	rD.	rA	rD.	STIBUDUUDE
OI CODE	10	١٨	טו	30BOLCODE

rB: radix

12/12

_	0x0							0xE	300
0x0									
0x10				Х ₀	X ₁	X ₂	X ₃	X ₄	
0x20	X ₅	Х ₆	Х7						
0x30									
0x40			Y ₀	Y ₄	Y ₆	Y ₂	Y ₁	Y ₅	
0x50	Y ₃	Y ₇							
0x60									

X ₀	X ₄	Х ₆	X ₂	R1
X ₁	X ₅	X ₃	X ₇	R2
				R3
Y ₀	Y ₁	Y ₂	Yz	R4
Y ₄	Y ₅	Y ₆	Y ₇	R5

FIG. 35